



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/065,793	11/19/2002	Jan Phillippe Eiras	1416.01	4753
21901	7590	12/13/2007	EXAMINER	
SMITH HOPEN, PA 180 PINE AVENUE NORTH OLDSMAR, FL 34677			EL CHANTL HUSSEIN A	
			ART UNIT	PAPER NUMBER
			2157	
			MAIL DATE	DELIVERY MODE
			12/13/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

MAILED

DEC 13 2007

Technology Center 2100

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/065,793
Filing Date: November 19, 2002
Appellant(s): EIRAS ET AL.

Molly L. Sauter
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed March 22, 2006 appealing from the Office action mailed Feb. 22, 2005.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Hite et al., U .S. Patent No. 6,763,040

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-30 are rejected under 35 U.S.C. 102(e) as being anticipated by Hite et al., U.S. Patent No. 6,763,040 (referred to hereafter as Hite).

As to claim 1, Hite teaches a computer program product for program level message traffic interception comprising:

a computer-readable medium (see col. 6 lines 4-47);

a protocol independent API core module stored on the medium, the API core module having an array of predetermined rules for intercepted message traffic (see col. 6 lines 14-67 and fig. 3-5, software emulator receives messages and parses the messages to determine a destination and status of devices); and

an interface communication emulator module communicatively coupling protocol-specific program level message traffic to the API core (see 6 lines 14-67, fig. 3-5, protocol converter 92 and CAN transport protocol client 94 are interpreted to be “an interface communication emulator module” that receives protocol messages from the CAN server and is forwarded to the API 110).

As to claim 2, Hite teaches the computer program product of claim 1 further comprising a message database communicatively coupled to the API core module, the message database further comprising an array of message properties for each message (see col. 11 lines 35-col. 12 lines 55 and fig. 11-13, messages has a format as shown in fig. 11-13 where TABLE A shows examples of the packet properties).

As to claim 3, Hite teaches the computer program product of claim 2 wherein the array of message properties further comprise message interpretation data (see col. 11 lines 35-col. 12 lines 55 and fig. 11-13 and TABLE A).

Art Unit: 2157

As to claim 4, Hite teaches the computer program product of claim 2 wherein the array of message properties further comprise message formatting data (see col. 11 lines 35-col. 12 lines 55 and fig. 11-13 and TABLE A).

As to claim 5, Hite teaches the computer program product of claim 2 wherein the array of message properties further comprise message routing data (see col. 11 lines 35-col. 12 lines 55 and fig. 11-13 and TABLE A).

As to claim 6, Hite teaches the computer program product of claim 2 wherein the array of message properties further comprise message default values (see col. 11 lines 35-col. 12 lines 55 and fig. 11-13 and TABLE A).

As to claim 7, Hite teaches the computer program product of claim 2 wherein the array of message properties further comprise message transmission rules (see col. 11 lines 35-col. 12 lines 55 and fig. 11-13 and TABLE A).

As to claim 8, Hite teaches the computer program product of claim 2 wherein the array of message properties further comprise enable-lockout combination data (see col. 11 lines 35-col. 12 lines 55 and fig. 11-13 and TABLE A).

As to claim 9, Hite teaches the computer program product of claim 2 wherein the array of message properties further comprise limits on message field values (see col. 11 lines 35-col. 12 lines 55 and fig. 11-13 and TABLE A).

As to claim 10, Hite teaches the computer program product of claim 2 wherein the array of message properties further comprise message field units (see col. 11 lines 35-col. 12 lines 55 and fig. 11-13 and TABLE A).

As to claim 11, Hite teaches the computer program product of claim 2 wherein the array of message properties further comprise user-defined identifiers (see col. 11 lines 35-col. 12 lines 55 and fig. 11-13 and TABLE A).

As to claim 12, Hite teaches the computer program product of claim 2 wherein the array of message properties further comprise interface information (see col. 11 lines 35-col. 12 lines 55 and fig. 11-13 and TABLE A).

As to claim 13, Hite teaches the computer program product of claim 2 further comprising a scenario module communicatively coupled to the message database, the scenario module further comprising state machine emulation definition, the definition providing event driven parameters responsive to message traffic (see col. 11 lines 19-35 and fig. 10).

As to claim 14, Hite teaches the computer program product of claim 13 wherein the event-driven parameters discriminate between messages based on message identification (see col. 11 lines 35-col. 12 lines 55 and fig. 11-13).

As to claim 15, Hite teaches the computer program product of claim 13 wherein the event-driven parameters discriminate between messages based on message contents (see col. 11 lines 35-col. 12 lines 55 and fig. 11-13 and TABLE A).

As to claim 16, Hite teaches the computer program product of claim 13 wherein the event-driven parameters discriminate between messages based on message occurrence (see col. 12 lines 31-61).

As to claim 17, Hite teaches the computer program product of claim 13 wherein the event-driven parameters discriminate between messages based on message frequency (see col. 12 lines 31-61 and col. 5 lines 65-col. 6 lines 30).

As to claim 18, Hite teaches the computer program product of claim 13 wherein the event-driven parameters discriminate between messages based on a count of the number of times an event's parameters have been satisfied (see col. 33).

As to claim 19, Hite teaches the computer program product of claim 13 wherein the event-driven parameters discriminate between messages based on a comparison with variables (see col. 5 lines 65-col. 6 lines 30).

As to claim 20, Hite teaches the computer program product of claim 13 wherein an event defined by the event-driven parameters modify the contents of a message (see col. 41-42).

As to claim 21, Hite teaches the computer program product of claim 13 wherein an event defined by the event-driven parameters route a message (see col. 11 lines 35-col. 12 lines 55 and fig. 11-13).

As to claim 22, Hite teaches the computer program product of claim 13 wherein an event defined by the event-driven parameters delete a message (see col. 41-42).

As to claim 23, Hite teaches the computer program product of claim 13 wherein an event defined by the event-driven parameters controls other events (see col. 11 lines 35-col. 12 lines 55 and fig. 11-13).

As to claim 24, Hite teaches the computer program product of claim 13 wherein an event defined by the event-driven parameters performs calculations (see col. 12 lines 31-61).

As to claim 25, Hite teaches the computer program product of claim 13 wherein an event defined by the event-driven parameters controls user displays (see col. 12 lines 31-61).

As to claim 26, Hite teaches the computer program product of claim 13 wherein an event defined by the event-driven parameters extracts at least one value from a message (see col. 12 lines 31-61).

As to claim 27, Hite teaches the computer program product of claim 13 wherein an event defined by the event-driven parameters creates and sends an arbitrary message defined in the database (see col. 12 lines 31-61).

As to claim 28, Hite teaches the computer program product of claim 13 wherein an event defined by the event-driven parameters transforms an incoming message into a different message defined in the database (see col. 11 lines 35-col. 12 lines 55 and fig. 11-13).

As to claim 29, Hite teaches the computer program of claim 13 wherein the actions triggered by an event provide logical branching, looping, iteration, and internal or external subroutine calling capability (see col. 11-col. 12).

As to claim 30, Hite teaches the computer program product of claim 13 wherein the communications interface emulator module is communicatively coupled to the scenario execution module which is communicatively coupled to the message database

whereby messages are received, reformatted into a message database compliant structure and outbound messages generated by the scenario module are passed back to the communications interface emulator module for protocol-specific transmissions

the computer program product of claim 13 further comprising a post-test data analysis capability wherein recorded data may be analyzed, abstracted, and displayed in a variety of text and graphical formats (see col. 11 lines 35-col. 12 lines 55 and fig. 11-13 and col. 12 lines 31-61).

(10) Response to Argument

As per appellants arguments filed on March 22, 2006, the appellant argues that Hite does not disclose a protocol-independent API (see Brief pages 9 lines 25-page 10 lines 15, argument A).

In reply to A) Hite teaches a system and method for sending and receiving control messages from a master controller to internet appliances over a network (see col. 2 lines 42-col. 3 lines 10). Fig. 2 shows the control messages are sent from a can browser 23 through an IA server 14 to the control network server 40 over a network (see col. 6 lines 19-43). The IA or internet appliance server includes a software device emulator 90 interpreted to be "API core". IA server also includes protocol converter 92 and CAN transport protocol client 94 interpreted to be "interface communication emulator module" (see fig. 4 and 5).

The software device emulator 90 consists of a device API that sends and receives messages from the web server and also sends and receives messages from the control network server (see col. 6 lines 44-67). The software device emulator 90

also includes a CAN input message processor 102 that receives input messages and parses fields of the message determine a message destination. Software device emulator also processes the messages to determine a current state of a software logical device and a next state in response to the processed message (see col. 6 lines 44-67). At least the determination of the destination of messages and the state of the logical device are interpreted by examiner to be "an array of predetermined rules".

Appellant argues that Hite does not disclose a protocol independent API since col. 1 and col. 11 states that the Hite reference teaches that messages received by the API contain protocol information. Examiner points out that the mere fact that the messages include protocol information does not make the API taught by Hite dependent on a specific protocol. Contrary to what the appellant argues, the IA server includes protocol converters 92 to convert messages between the CAN server and the client (see fig. 3 and 4 and col. 6 lines 19-32) and therefore the API taught by Hite is not dependent on a specific protocol since the software device core is capable of handling a plurality of protocols.

In addition, the claim language used by the applicant defines the claimed "protocol independent API" to have "an array of predetermined rules for intercepted message traffic". Protocol in the simplest definition is a *set of rules*. Therefore the claimed API is dependent on an array of rules which renders the claimed API dependent on some form of protocol. Even assuming that Hite's API processes protocol information, according to the definition of the claimed protocol independent API, Hite still teaches the scope of the claimed limitation.

The appellant argues that Hite does not disclose an array of predetermined rules (see Brief pages 9 lines 25-page 10 lines 15, argument B).

In reply to B) Hite teaches the software device emulator 90 also includes a CAN input message processor 102 that receives input messages and parses fields of the message determine a message destination. Software device emulator also processes the messages to determine a current state of a software logical device and a next state in response to the processed message (see col. 6 lines 44-67). At least the determination of the destination of messages and the state of the logical device are interpreted by examiner to be “an array of predetermined rules”. Claim language does not specify what the contents of the array of rules are and therefore examiner broadly interprets the parsing of the message and determining the destination and the state of the device to be the array of predetermined rules.

The appellant argues that Hite does not disclose an interface communication emulator module communicatively coupling protocol-specific message traffic to the API core (see Brief pages 10 lines 12-26, argument C)

In reply to C) Hite discloses protocol converter 92 and CAN transport protocol client 94. The CAN transport protocol client 94 receives messages from the CAN server. The protocol converter converts messages between the CAN server and the client (see fig. 3 and 4 and col. 6 lines 19-32). The messages are then forwarded to the software device emulator to be transmitted through the API to the web server (see fig. 3 and 4). Therefore protocol converter 92 and CAN transport protocol client 94 are interpreted to be “an interface communication emulator module” that receives protocol messages from

the CAN server and is forwarded to the API 110 and therefore "coupling the message traffic to the API".

The appellant argues that the finality of the office action mailed on Feb. 22, 2005 is premature (see Brief pages 8 lines 1-pages 9 lines 15, argument D).

In reply to D) Applicant amended claim 1 in the response filed on Oct. 28, 2004 to include the limitation program level message which changed the scope of the claim. The change in scope necessitated the new grounds of rejection and therefore the action was made final in accordance with MPEP § 706.07(a).

In addition, the finality of the rejection is not an appealable matter and therefore appellant's argument is improper.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,



ZARNI MAUNG
SUPERVISORY PATENT EXAMINER

Conferees:

Hussein El-chanti

June 8, 2006



ARIO ETIENNE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100